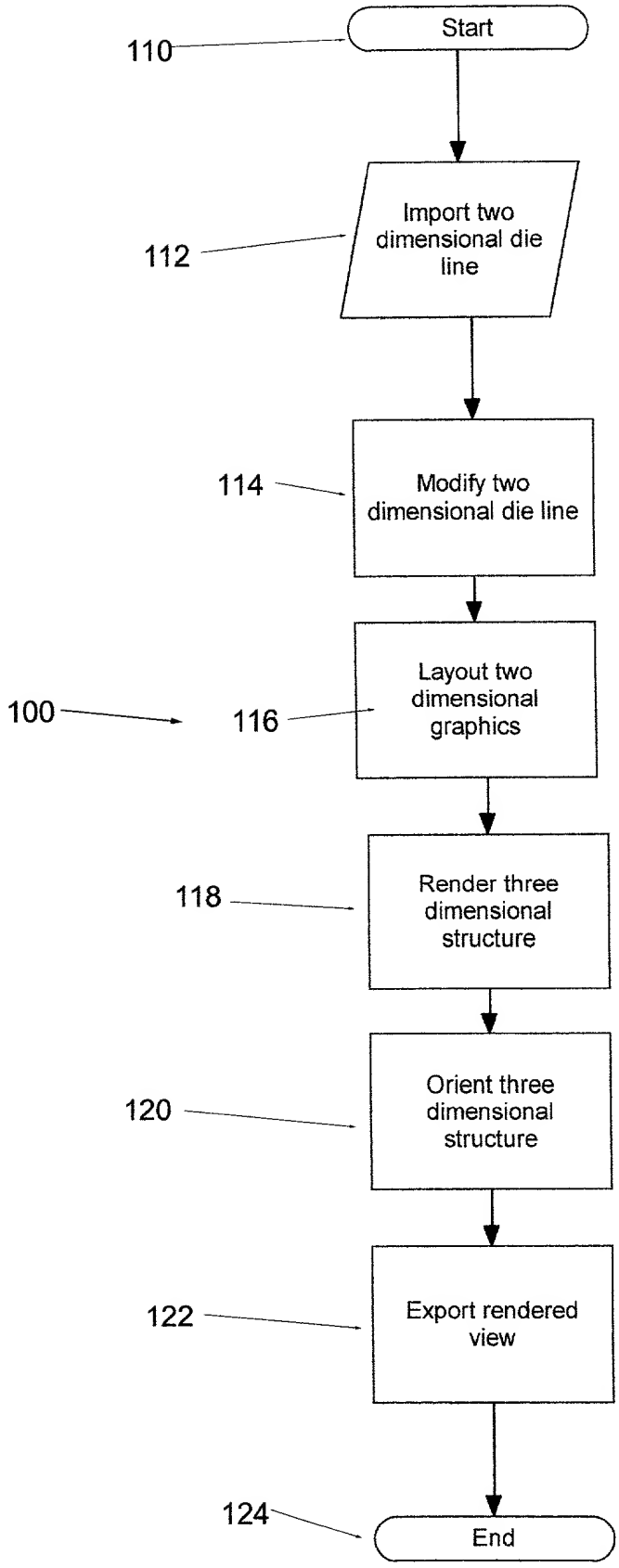
Start
110
Import two dimensional die line
112
Modify two dimensional die line
114
Layout two dimensional graphics
100
116
Render three dimensional structure
118
Orient three dimensional structure
120
Export rendered view
122
End
124

Figure 1

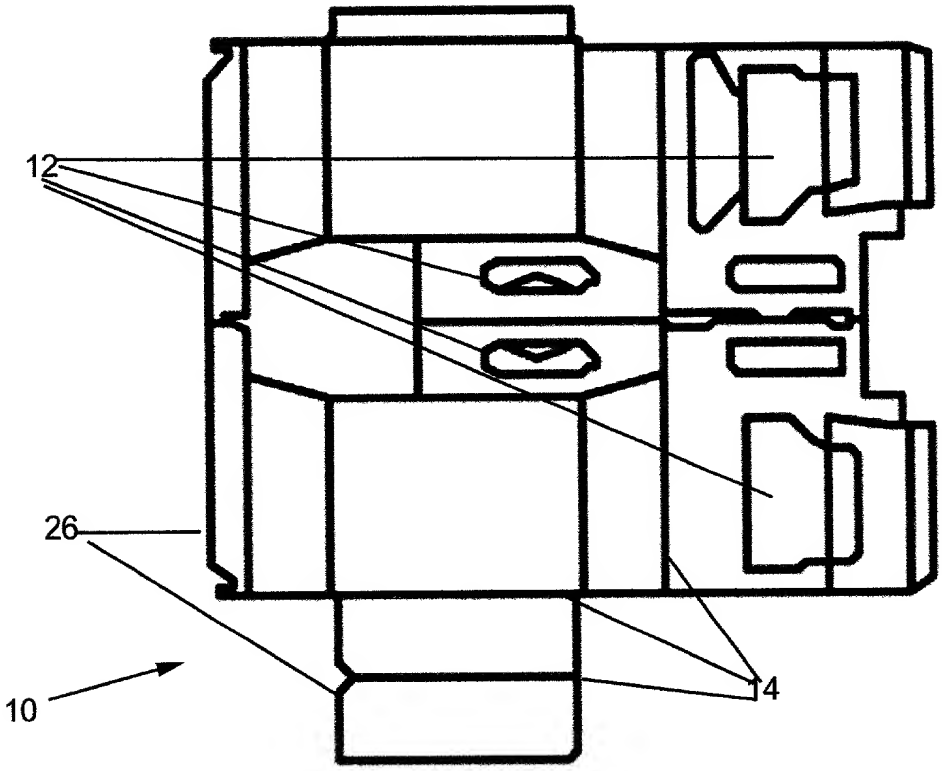

Figure 2

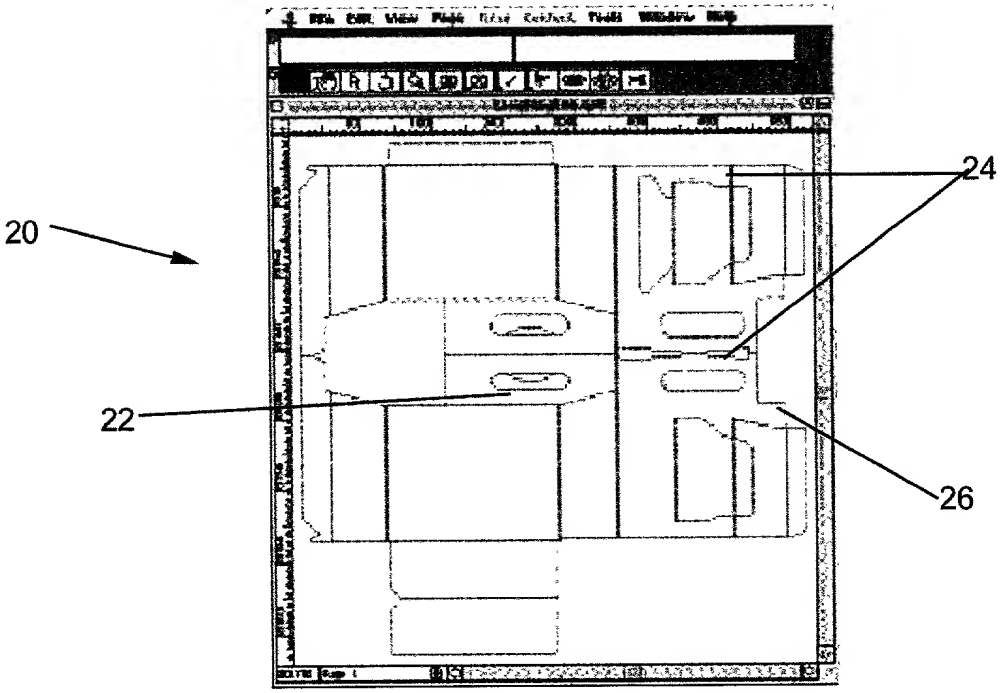

Figure 3

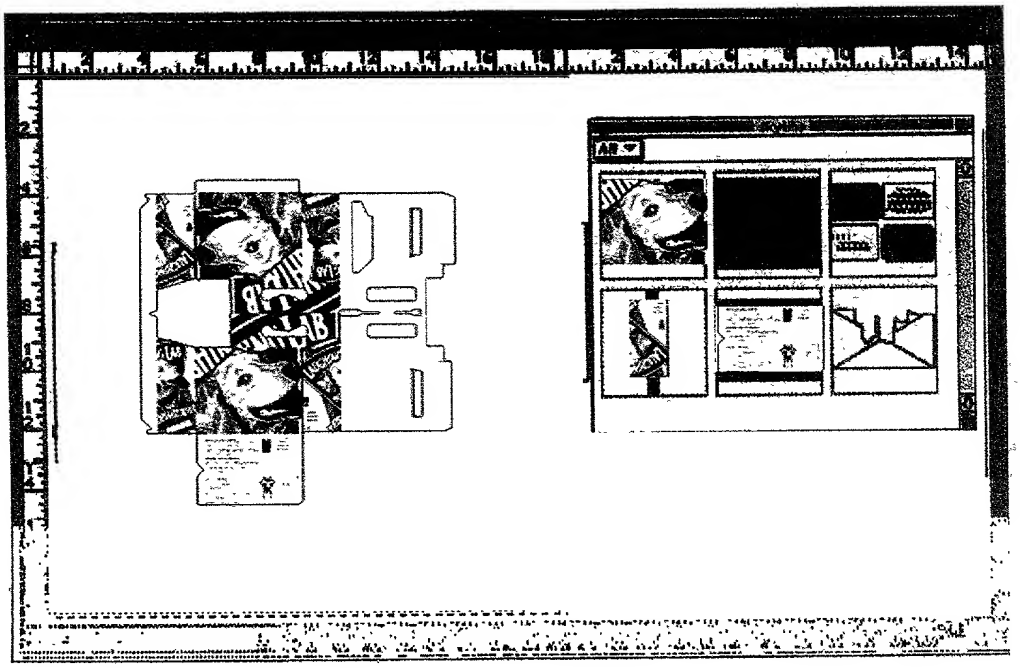

Figure 4

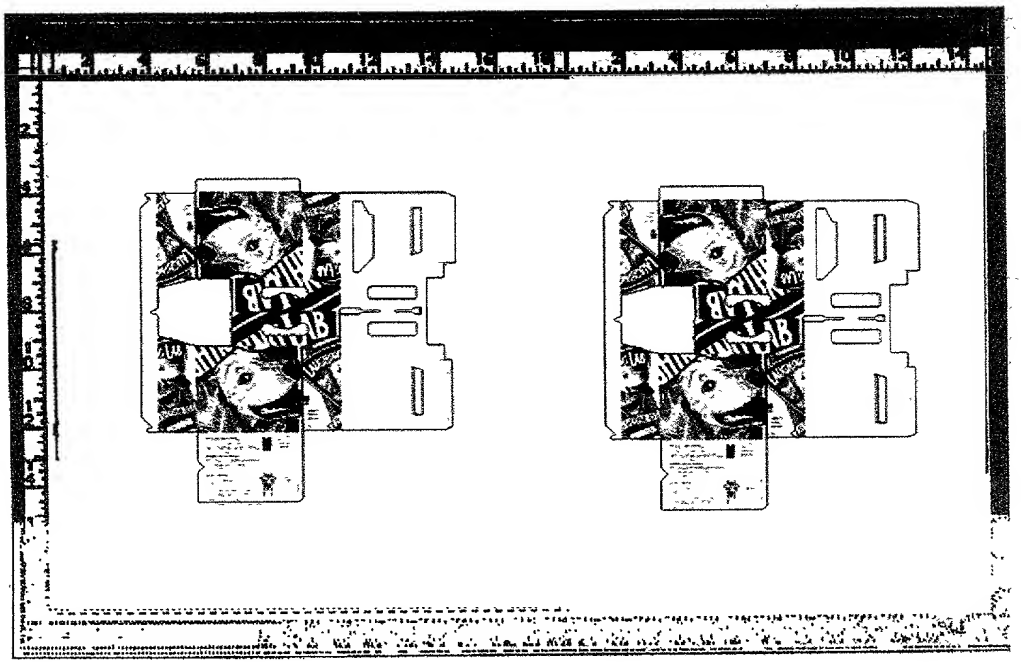

Figure 5

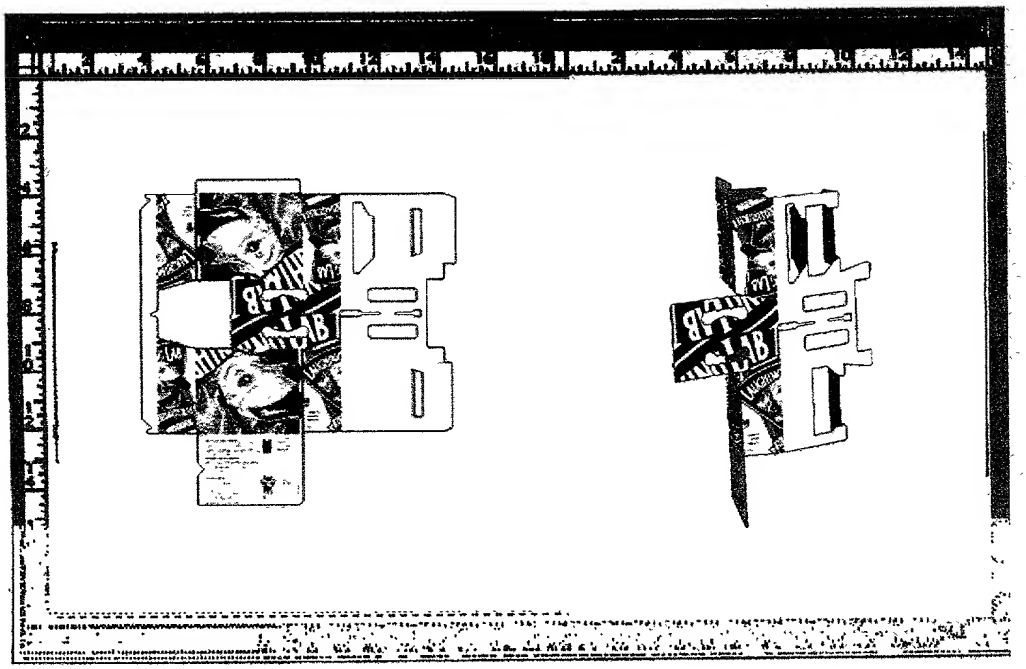

Figure 6

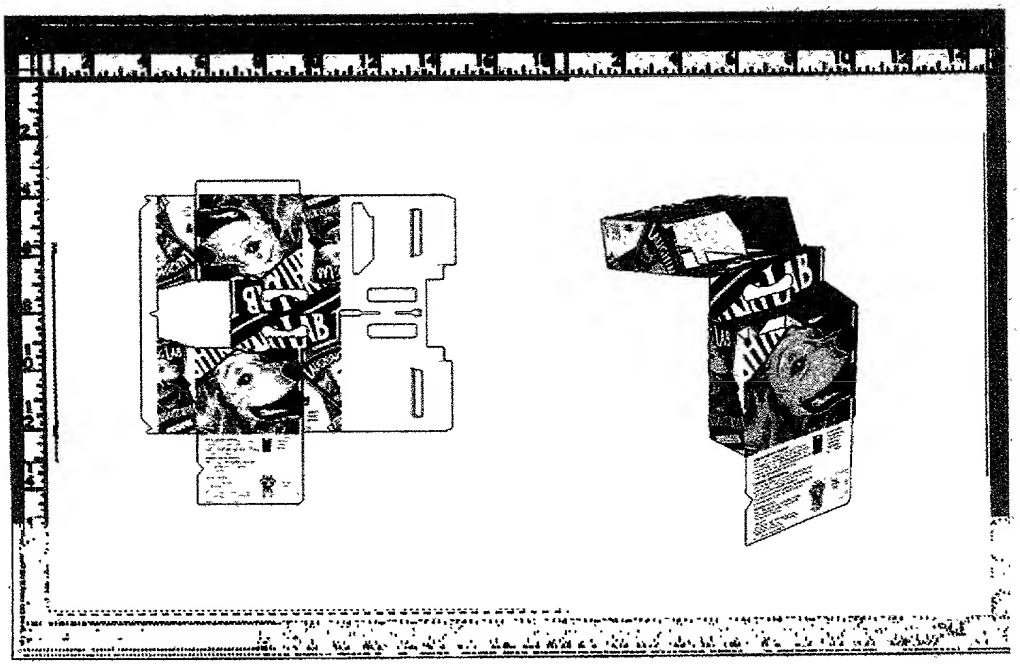

Figure 7

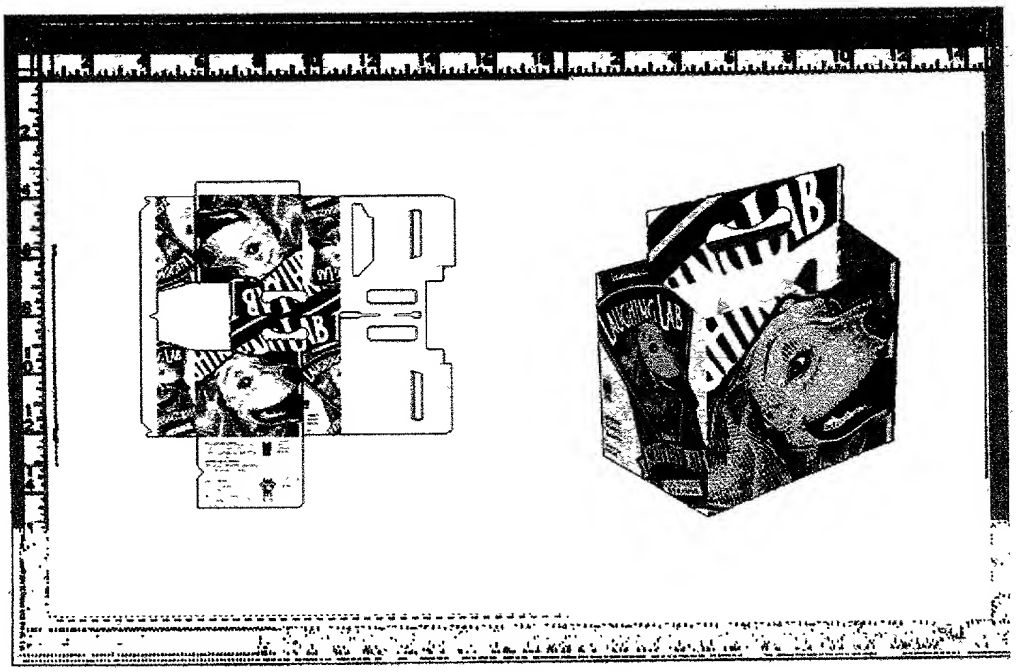

Figure 8

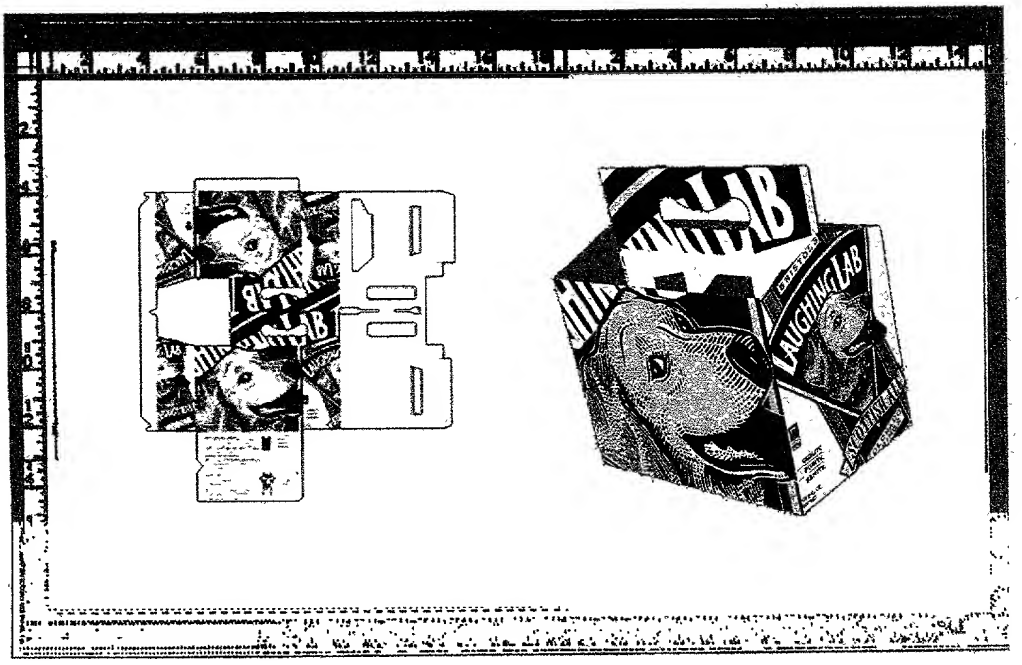

Figure 9

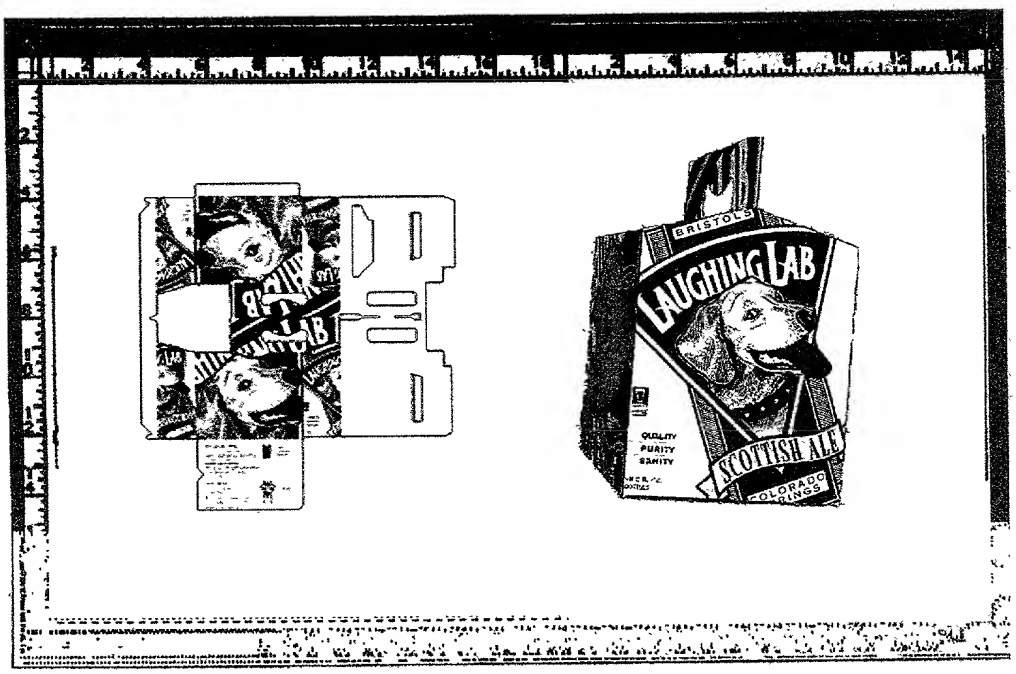

Figure 10

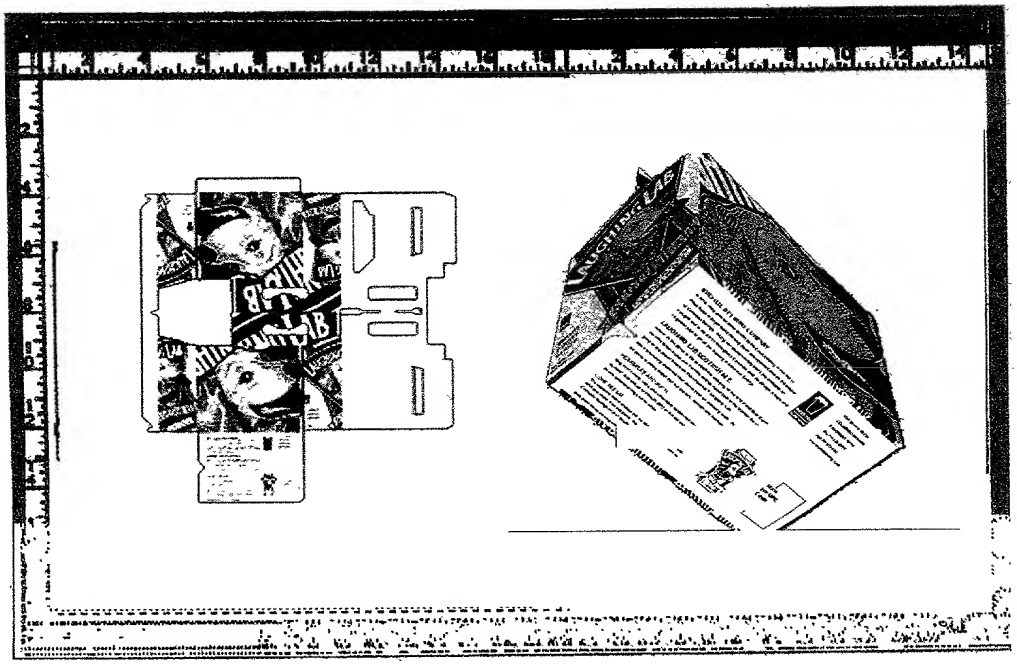

Figure 11

Start

Validate the two-dimensional die line
If not valid then alert user
If valid, then:
Copy the x,y coordinates and edge information from the two-dimensional die line to a two dimensional surface drawing
Enhance the two-dimensional surface with additional structure to account for barrels about crease lines
Triangulate the two-dimensional surface, associating each triangle with a single rigid panel within the two-dimensional surface
Record the coordinates in two-dimensional graphics space for each point on the two-dimensional surface
Copy the two-dimensional surface to a flattened three-dimensional structure
Fold the three-dimensional structure by generating and then applying fold transforms for each rigid panel along every crease line
Add an inside surface and outer edges to the three-dimensional structure in camera space, accounting for relative orientations and perspectives
Retrieve the two-dimensional graphics from the document as a texture bitmap
Clear the view and distance buffers
Compute the x,y,z coordinates of the three-dimensional structure in camera space, accounting for relative orientations and perspectives
Clip any portions of the three-dimensional structure in camera space that extend beyond the bounds of the view buffer
Iterate across all triangles comprising the clipped three-dimensional structure in camera space
    Construct a lighting intensity map for each triangle
    Iterate across each scan line intersecting each triangle
        Iterate across all sample points within the scan line within the triangle
            Compute the distance z from the triangle to the camera at each point
            If no other surface point is shown as closer to the camera in the distance buffer
                Compute the coordinates in two-dimensional graphics space for this point
                Retrieve the r,g,b components from the texture bitmap at these coordinates
                Retrieve the lighting intensity for this point from the lighting intensity map
                Modify these r,g,b values to account for lighting intensity
                Store the modified r,g,b components in the view buffer
                Store the z component in the distance buffer
            End if
        End iteration
    End iteration
End iteration
Filter and then copy the view buffer to the main frame buffer of the display device
Else (if the two-dimensional die line is not Valid):
    Report an appropriate error to the user
End else

End

Figure 12